A Simple View of Context-Based Robot Intelligence

Most hobby robots already change their behaviour according to context.

For example,

- slow down when the battery is low,
- avoid obstacles when moving quickly,
- use different behaviour during the day and at night,
- become more cautious when operating near people.

Usually these behaviours are programmed using state machines, rules or special-case code.

But there is another way to think about intelligent behaviour.

Instead of building one large complicated controller, we can build a robot from a **simple sequence of context-selected linear mappings.**

This idea is called **Concatenated Context Selected Linear Mappings (CCSLMs).**

The Basic Building Block

Each stage in the computation performs exactly three operations.

1. Examine one or more context variables.
2. Select one linear mapping.
3. Apply that linear mapping and pass the result to the next stage.

That is all.

There are no branches, no alternative execution paths and no complicated routing.

The computation simply moves through a chain of stages, one after another.

Each stage selects a local linear mapping based upon its own context.

Every Stage Can Have Its Own Context

An important feature is that every stage can use different context variables.

For example,

Stage 1

Context: camera image

Purpose: visual interpretation

Stage 2

Context: time of day

Purpose: adjust daytime and nighttime behaviour

Stage 3

Context: robot speed

Purpose: choose different control behaviour when stopped, moving slowly, cruising or moving quickly

Stage 4

Context: battery level

Purpose: conserve energy as the battery becomes depleted

Stage 5

Context: internal personality variables

Purpose: influence exploration, caution or sociability

Each stage remains extremely simple.

Only the context used to select its linear mapping changes.

A Different View of ReLU Networks

Interestingly, ordinary ReLU neural networks can be viewed in a similar way.

A ReLU layer may be written as

$$y = W D(x) x$$

where $D(x)$ is a diagonal decision matrix whose entries are either 0 or 1.

The decision matrix selects which local linear mapping is active.

Each layer therefore behaves as a context-selected linear mapping.

A ReLU network can be viewed as a sequence of these simple stages connected together.

CCSLMs extend this idea by allowing the context to come from many sources, not just the current input.

Richer Contexts

The context used by a stage might include

- camera images,
- microphone input,
- time of day,
- battery level,
- robot speed,
- mission status,
- current task,
- previous actions,
- internal memory,
- user commands,
- personality variables.

Different stages can specialise in different aspects of the robot's behaviour.

Personality as Context

One particularly interesting possibility is to include long-lived internal variables.

Examples include

- curiosity,
- confidence,
- patience,
- caution,
- friendliness,
- energy.

These variables change slowly over time.

Instead of directly controlling the robot, they simply influence which linear mappings are selected at different stages.

In this way, the robot's personality becomes part of its computational context.

Why This Is Attractive

The overall architecture remains remarkably simple.

Each stage only performs one local linear computation.

The intelligence comes from selecting the appropriate computation for the current context.

This separation makes the system easier to understand and potentially easier to design than one large monolithic controller.

A hobbyist can introduce new behaviours simply by adding new context variables or refining the context selection for a particular stage, without redesigning the entire computation.

Training

The linear mappings may be designed manually, estimated from data, or learned automatically.

Because each stage performs only a linear computation while context selection is handled separately, the architecture may admit simpler training procedures than those used for conventional nonlinear networks. Standard ideas from backpropagation still apply, but the mathematical structure of CCSLMs may allow alternative formulations that take advantage of the piecewise-linear nature of the computation.

Looking Ahead

CCSLMs suggest a different way to think about robot intelligence.

Rather than viewing intelligence as one enormous nonlinear program, they describe it as a simple sequence of context-selected linear mappings.

Each stage asks a single question:

> "Given my current context, which linear mapping should I apply?"

Everything else follows from repeatedly applying that simple idea.

For hobby robotics, this offers an appealing combination of simplicity, modularity and flexibility. A robot can respond not only to what it currently senses, but also to time, memory, internal state and even personality variables, while the underlying computation remains a straightforward chain of context-selected linear operations.